

TX522B

应用指南

Application Note

AN01020202 REV1.04 Date: 2010/05/27

关键词

- TX522B 读卡模块
- Mifare 1 S50, S70
- 扇区 密钥
- 一卡多用 一卡通
- 非接触式 智能卡
- 自动寻卡

摘 要

TX522 系列模块是基于 13.56MHz 频率的 Mifare 卡读写模块，符合 ISO14443A 标准，可支持 Mifare1 S50、Mifare1 S70、Mifare Light、Mifare UltraLight、Mifare Pro。TX522 系列 Mifare 读写模块具有易用、高可靠、多种接口、体积小等特点，可帮助用户方便、快捷地将当今最流行的非接触式 IC 卡技术融入到系统中，提高产品的档次。

TX522B 能够配置为自动寻卡模块，上电后无需向模块发送任何命令，只要有卡靠近模块就能主动通过 uart 发送卡号。

本文详细介绍了 TX522B 模块的基于 51 单片机的 C51 库函数的使用方法。

目　录

1. 适用范围........4
2. TX522B 简介........5
2.1 引脚描述........5
2.1.1 天线一体化模块引脚描述........5
2.1.2 天线分体模块引脚描述........6
2.2 典型电路........6
2.3 技术参数........7
2.4 极限参数........7
2.5 直流特性........7
2.6 封装及机械尺寸........8
3. 只读卡号的最简单应用........10
3.1 硬件配置........10
3.2 检测到卡片指示........10
3.3 返回卡号说明........10
4. TX522B 读卡模块数据传输协议........11
4.1 串口协议........11
4.2 控制字符定义........11
4.3 协议描述........11
4.3.1 数据通信帧描述........11
4.3.2 主机发送命令至模块........12
4.3.3 从机返回数据给主机........12
5. TX522B 的 C51 函数........13
5.1 函数列表........13
5.2 函数返回状态值列表........14
5.3 函数描述........14
5.3.1 配置—Config........15
5.3.2 配置—ConfigAnt........15
5.3.3 关闭一Close........16
5.3.4 获取信息—Get Info........17
5.3.5 装载密钥—Load_Key........18
5.3.6 激活卡片并获取卡号—Get_CardSnr........19
5.3.7 带验证的写—Write_Auth........20

5.3.8 写 UltraLight—Ul_Write 21
5.3.9 带验证的读—Read_Auth......22
5.3.10 带验证的写值块操作—WriteValue_Auth......23
5.3.11 带验证的值块操作—Value_Auth......24
5.3.12 带验证的读值块操作—ReadValue_Auth......25
5.3.13 暂停—Halt......26
5.3.14 复位—Reset......27
5.3.15 自动寻卡—Auto_Detect 28
5.3.16 读取自动检测数据—Read_ADDT......30
5.3.17 请求—Request......31
5.3.18 带层级设置的防碰撞—Casc_Anticoll 33
5.3.19 带层级设置的选择—Casc_Select 35
5.3.20 证实 2—Auth2......36
5.3.21 直接密码证实—Auth_Key 37
5.3.22 写—Write......38
5.3.23 读—Read 39
5.3.24 带内部自动传送的值操作 1—Value......40
5.3.25 置位控制位—Set_Control_Bit......41
5.3.26 清除控制位—Clr_Control_Bit......41
5.3.27 输出蜂鸣器信号—Buzzer......42
6. 使用串口调试助手开发调试 TX522B 模块......44
6.1 软硬件准备......44
6.2 电路连接......44
6.3 设置串口调试软件......44
6.4 向 TX522BT+发送命令 45
7. Mifare 卡工作状态介绍......47
8. 免责声明......48
9. 修订历史......49
10. 销售信息......49

1. 适用范围

本文对 TX522B Mifare 卡读写模块在 51 单片机下的 C51 库函数做了非常详细的讲解，使用 51 系列单片机与 TX522B 进行通讯的 TX522B 用户参考本文提供的各函数，可以非常容易的编写操作 Mifare 卡的应用程序。不使用 51 系列单片机，而使用其它处理器的 TX522B 用户，也可以参考本文所描述提供的通讯格式和函数，将本文提供的函数很容易的移植到其它处理器上。

对于 C51 函数的调用，用户可不必关心数据块格式，只要理解函数的功能，输入、输出参数即可。当用户自己编写函数（包括非 C51 下的函数）时，就需了解数据块格式，必须按照数据块的格式来编写函数。

2. TX522B 简介

TX522B 是一个简单的串行读写模块，不带后缀“＋”时，接口为 CMOS 电平，用于与常用的微处理器（如单片机、ARM）的 UART 接口；当采用带后缀“＋”时，TX522B＋模块内部带有 RS232 电平转换电路，能够直接与 PC 机的串口连接通信。

用户通过主机（包括单片机、ARM、DSP 以及 PC 机等）向 TX522B 模块发送命令来对 TX522B 进行读写控制。本应用指南将重点描述 TX522B 与主机之间串行通信协议和命令。

2.1 引脚描述

2.1.1 天线一体化模块引脚描述

表 1 外接天线接口 J1

接　口	管　脚	符　号	IO 类型	功能描述
J1(1)	J1-1	TX1	输出/O	天线发送端 1
	J1-2	GND	地/Power	地
	J1-3	TX2	输出/O	天线发送端 2
	J1-4	GND	地/Power	地

(1) J1 为模块与天线的接口,对于天线一体化（带后缀 T）的模块，如果用户使用模块上的 PCB 印制天线，则可以不使用该接口；对于非天线一体化（不带后缀 T）的模块，用户要通过该接口来连接天线。

表 2 与用户 MCU 接口 J2

接 口	管脚	符 号	IO 类型	功能描述	上电后的状态
J2(1)	J2-1	CTRL	输出/O	控制信号输出	1
	J2-2	BZ	输出/O	外部蜂鸣器驱动电路控制信号，需要串电阻	1
	J2-3	INT_OUT	输出/O	自动寻卡中断输出，低电平有效	1
	J2-4	VCC	电源/Power	电源正极	
	J2-5	NC		空闲脚	
	J2-6	GND	地/Power	地	
	J2-7	RXD	输入/I	UART 接收端	1
	J2-8	TXD	输出/O	UART 发送端	1

(1) J2 为模块与用户控制器的接口。

(2) 可将自动寻卡配置为寻到卡后自动回发，然后用 UART 接收中断，就可以不使用 INT_OUT 管脚。但使用 UART 接收中断而不使用中断管脚时，用户程序中的关中断时间不能大于 20ms，因为 TX522B 模块在寻到卡后，主动向主机发送请求 STX，如果在 20ms 内得不到应答就直接发送数据。

表 3 用户接口 J3

接 口	管 脚	符号	IO 类型	功能描述
J3	J3-0	+3.3V	地/Power	3.3V 电源输出，最大提供 70mA 电流
	J3-1	NC	-	预留未来使用
	J3-2	NC	-	预留未来使用
	J3-3	NC	-	预留未来使用
	J3-4	AutoDet_en	输入	上电主动回发卡号使能管脚，和 GND 短接时使能，悬空时为 1
	J3-5	GND	地/Power	地

2.1.2 天线分体模块引脚描述

表 4 分体式 TX522B 引脚描述

管 脚	符 号	IO 类型	功能描述	上电后的状态
1	TX1	输出/O	天线发送端 1，同 25 脚，内部相连	
2～10	GND	地/Power	地	
11	AutoDet_en	输入	上电主动回发卡号使能管脚，和 GND 短接时使能，悬空时为 1	1
12	CTRL	输出/O	控制信号输出	1
13	BZ	输出/O	外部蜂鸣器驱动电路控制信号，需要串电阻	1
14	NC		空闲脚	
15	TXD	输出/O	UART 发送端	1
16	VCC	电源/Power	电源正极，请外接 100uF 电解电容	
17	GND	地/Power	地	
18	RXD	输入/I	UART 接收端	1
19	INT_OUT	输出/O	自动寻卡中断输出，低电平有效	1
20~22	NC		空闲脚	
23，24	GND	地/Power	地	
25	TX1	输出/O	天线发送端 1，同 1 脚，内部相连	
26，27	NC		空闲脚	
28	+3.3V	地/Power	3.3V 电源输出，最大提供 70mA 电流	

2.2 典型电路

TX522BT 模块可以与任何带有 UART 口的 MCU 接口，图 1 所示为 TX522BT 与 MCS51 单片机 UART 的典型接口。

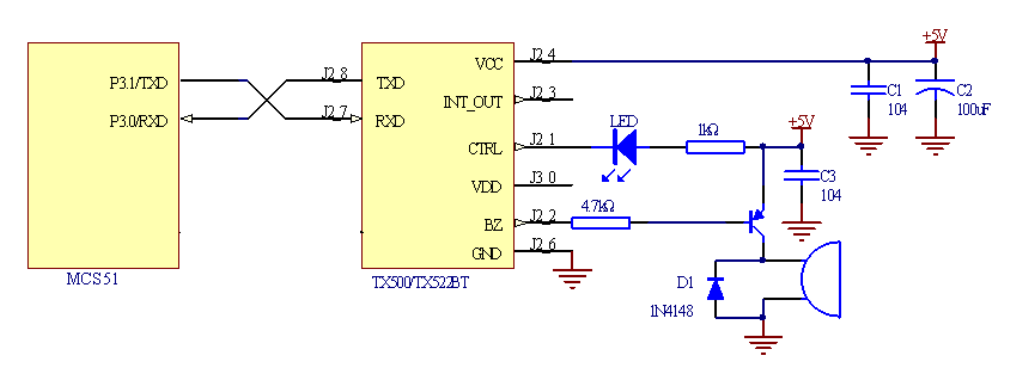


图 1 TX522BT 与 MCS51 单片机接口图

当采用带 RS232 电平转换的 TX522BT+模块时，如 图 2 所示，读写模块可以通过串口电缆直接与计算机相连，通过计算机来控制 TX522BT+。

图 2 TX522BT+与 PC 机串口接口图

2.3 技术参数

表 5 TX522 模块技术参数表

产品型号	TX522 系列
功　耗	25 毫安/直流 5V；
工作频率	13.56 兆赫兹
读卡距离	50～100 毫米（mifare1 卡）
接口方式	UART、RS232
数据传输速率	UART/RS232：9600～230400bit/s
支持卡类型	mifare1 S50、mifare1 S70、mifare UltraLight、mifare Pro
尺　寸	分体式（不带后缀 T）：37mm×18.5mm×6mm
	天线一体化（带后缀 T）：58mm×34.5mm×2.8mm

2.4 极限参数

每个管脚的对地电压 ……………………………………………………-0.5～+5.5V
Vcc 对地的电压 ……………………………………………………-0.3～+7.0V
每个管脚的最大 I_{OL} ……………………………………………………20mA
湿度（相对湿度）……………………………………………………5%～95

超出“绝对最大额定值”列出的值的条件下工作会造成器件的永久损坏。以上列出的是器件正常工作的额定值，并未涉及器件在这些条件或超出这些条件下的功能操作。器件不能长时间工作在绝对最大额定值条件下，否则会影响其可靠性。

2.5 直流特性

VCC=+5.0V，器件都工作在建议的温度范围-30～85℃条件下，除非特别说明。

表 6　TX522B 模块的直流特性

符 号	参　数	测试条件	最小值	典型值(1)	最大值	单位
VCC	工作电压		+4	+5.0	+7	V
VDD	输出电压			3.3V		V
I_{VDD} max	VDD最大输出电流				70	mA
T_{OK}	上电后稳定工作时间		5			ms
I_{CC}	电流消耗	读卡芯片配置成功		25		mA
V_{IL}	输入低电平		0.7	1.3	-	V
V_{IH}	输入高电平		-	2	2.3	V
V_{OL}	输出低电平	I_{OL}=20mA		0.6	1.0	V

		I_{OL}=3.2mA		0.2	0.3	V
V_{OH}	输出高电平	I_{OH}=-20uA	3	3.1		V
I_{IL}	逻辑低电平输入电流	Vpin=0.4V			-80	μA
I_{TL}	逻辑1到0变化电流	Vpin=2V	-30		-450	μA
I_{OL}	低电平时的灌电流				-20	mA
I_{OH}	高电平时的拉电流				20	μA
C_{IO}	管脚输入电容				15	pF
T_{OP}	工作温度(I)		-30		+85	℃
T_{STR}	存储温度		-55		+125	℃

(1) 典型值是难以保证的，这个值是在常温条件下测试得到。

(2) 模块上电后，必须等待 5ms 以上时间才能稳定工作。

2.6 封装及机械尺寸

TX522 模块根据是否采用天线一体化有两种封装型号：TX522 和 TX522T。天线分体式 TX522 封装尺寸如图 3 所示，天线分体式 TX522 封装尺寸如图 4 所示。为保证用户目标板和 TX522 系列模块接口尺寸一致，用户可以在 http://www.txrfid.com 上获取 TX522 系列模块元件库和 PCB 封装。

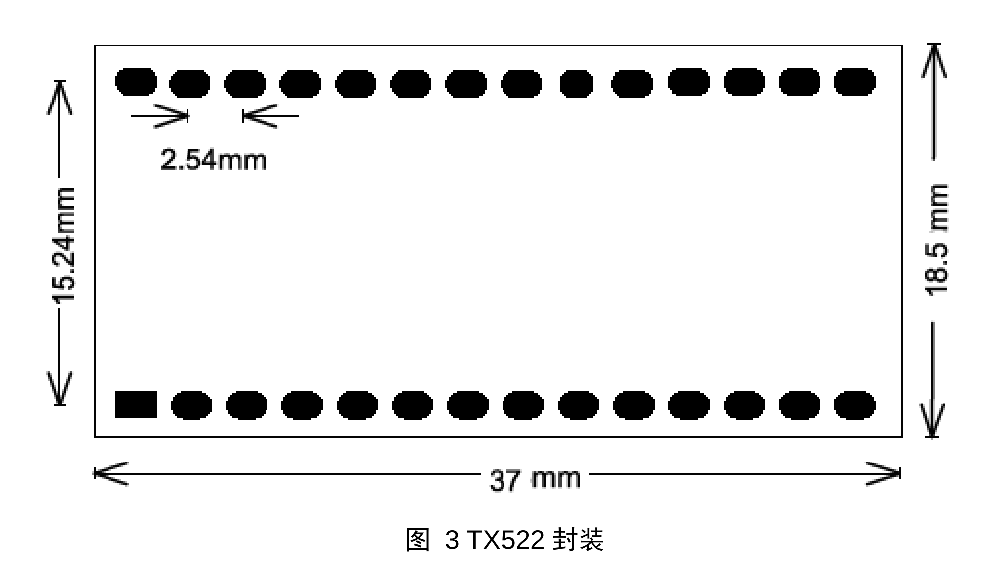


图 3 TX522 封装

单位: mm

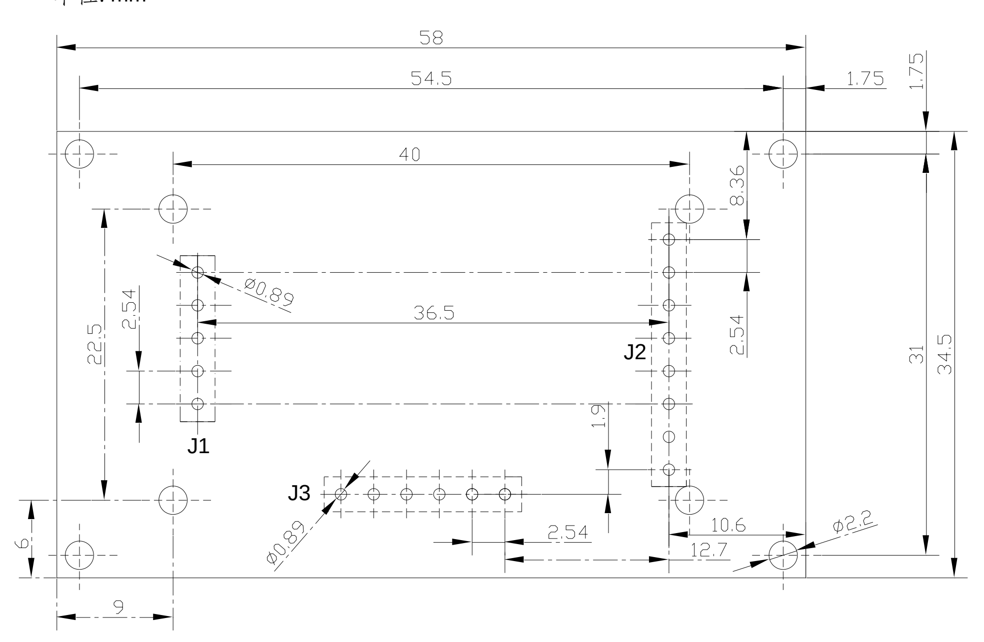


图 4 TX522T 封装

3. 只读卡号的最简单应用

如果您的应用只是要读取 mifare 的卡号，那您只需要关注本小节，本小节以后的内容您都都可以忽略。

TX522B 能够配置为上电自动寻卡返回卡号模式，上电后无需向模块发送任何命令，只要有卡靠近模块就能主动通过 uart 发送卡号。自动寻卡返回卡号模式操作按下面步骤实现。

3.1 硬件配置

1.将 J3_4 和 J3_5 短接，以使能上电启动自动寻卡并发送卡号功能。

2.按照图 6 所示连接 TX522T 和 PC 机的串口或其它主机串口。根据硬件不同可选带 RS232 转换和不带 RS232 转换。

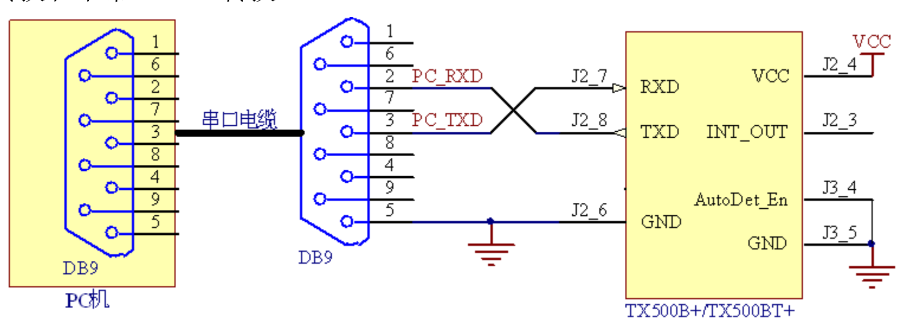


图 5 硬件接线图

3.将您接收所用串口配置为：9600 波特率，1 个起始位，8 个数据位、无奇偶校验位、1 个停止位。

3.2 检测到卡片指示

当有 mifare 卡接近 TX522BT 时，有卡的指示管脚 J2_3（INT_OUT）变为为低电平，在 J2_3 变为低电平 5ms 后，TX522BT 将卡号发出。如果用户使用串口接收中断方式进行卡号的接收，就可以不使用指示管脚 J2_3（INT_OUT）。使用中断接收要注意关中断时间不能超过 20ms。

一次刷卡，只发送一次卡号，如果卡片一直不拿开则不重复发生卡号，但有卡指示管脚仍然有效，即一直为低。

3.3 返回卡号说明

上电自动寻卡返回卡号模式下能读取所以 mifare 家族的卡号，包括 Mifare1 S50、Mifare1 S70、Mifare Light、Mifare UltraLight、Mifare Pro 等。

对于 4 字节卡号的 mifare 卡（如 mifare1 S50），返回的格式为 4 字节卡号+4 字节卡号的异或取反校验。卡号低字节在前。

如果是 7 字节卡号的 mifare 卡（如 Mifare UltraLight），返回的格式为 7 字节卡号+7 字节卡号的异或取反校验。

4. TX522B 读卡模块数据传输协议

4.1 串口协议

主机与 TX522B 串行接口通信过程中一帧的数据格式采用 1 个起始位，8 个数据位、无奇偶校验位、1 个停止位。

通信协议分为两步，第一步波特率设置，第二步进行数据通信。

用户可以通过串口调试助手来调试开发 TX522B，具体见 6 使用串口调试助手开发调试 TX522B 模块。

4.2 控制字符定义

下表列出了 TX522B 与主机串行通信过程中用到的控制字符定义。

表 7 TX522B 串行通信控制字符表

描述	定义	值
开始符	STX	0x20
终止符	ETX	0x03
应答	ACK	0x06
无应答	NAK	0x15

4.3 协议描述

4.3.1 数据通信帧描述

数据通信以一帧为单位进行，格式如下：

表 8 数据通信帧结构

起始符 STX	包号 SEQNR	命令/状态 CMD/Status	数据长度 Length	数据 DATA	校验和 BCC	帧结束符 ETX
1byte	1byte	1byte	1byte	N bytes	1byte	1byte

数据帧中各字段说明如表 9 所示：

表 9 数据帧各字段说明

字段	长度	说明	补充
STX	1	STX=0x20，数据帧的起始符，每一帧数据都是以 STX 开始	
SEQNR	1	该数据帧包序号，从 0 到 255 循环。可以用来作为通信间的错误检查，从机（模块）接收到主机发来的信息，在应答信息中发出一个同样的 SEQ 信息，主机可以通过此信息检查是否发生的“包丢失”的错误。第一个包的 SEQ 可为任意值。	该字段主机发送和接收的应该相同
Cmd /Status	1	主机——从机：命令 Command 从机——主机：状态 Status	
Length	1	该帧所带数据信息长度 若模块返回状态不为 0（OK），则 Length=0。	
DATA	Length	数据信息,长度等于 Length	
BCC	1	校验和。从包号（SEQNR）开始到数据（DATA）的最后一字节异或取反。	
ETX	1	ETX=0x03，是一个帧的结束标志	

为了下文的描述，把 SEQ+CMD/Status+LEN+DATA+BCC 同称为数据块 DATA Block。

表 10 数据块 DATA BLOCK

DATA BLOCK	包号 SEQNR	命令/状态 CMD/Status	数据长度 Length	数据 DATA	校验和 BCC
	1byte	1byte	1byte	N bytes	1byte

数据帧接收规则：

- 一帧的结束一定是 ETX，但接收到 0x03 则不一定是帧结束；
- 帧长必须不小于 7 字节，最大不能超过 64 字节，且帧长必须等于数据长度加 6；
- BCC 计算必须正确。

主机发送数据必须符合以上规则，否则从机不会执行任何命令，也不会有任何错误响应。同样主机接收从机的数据也必须符合以上规则，如果不符合，主机必须丢弃这帧数据。

4.3.2 主机发送命令至模块

除短接 J3_4 启动自动寻卡读取卡号以外，通信必须先由主机发送命令和数据给 TX522B，TX522B 执行命令完毕后，再将执行命令后的状态和响应数据发回给主机。

主机发送格式如表 11 所示。

表 11 主机发送给 TX522B 遵循的格式

主机	数据传送方向	TX522B	说明
STX+DATA BLOCK+ETX	→		一次将一帧数据全部发送完。

用户在给 TX522B 模块发送命令时，连续的发送 STX (0x20)+数据块+0x03(结束符)。通过判断 TX522B 返回数据的正确性来判断 TX522B 是否正确执行了本条命令。如果 TX522B 系列模块在接收到按照上面格式发送的命令后，会在接收到数据帧第一字节 STX 后给出应答 0x06，用户可以不用理会。

4.3.3 从机返回数据给主机

TX522B 在接收到主机发送的数据后，根据数据块的内容执行相应命令，并将执行命令后的状态或者数据（以下将状态和数据都统称为数据）返回给主机。

TX522B 发送格式如表 12 所示：

表 12 TX522B 发送格式

TX522B	数据传送方向	主机	说明
STX	→		若 TX522B 在 20ms 内未收到主机回应 ACK，直接发送 DATA BLOCK+ETX
	←	ACK	
DATA BLOCK+ETX	→		

TX522B 返回数据给主机时，首先发送起始符 STX 给主机，并等待主机返回应答信号 ACK。TX522B 在接收到主机返回的 ACK 时，会立即将需要返回的数据块发出。如果主机在 20ms 内一直不应答 ACK，那么 20ms 后，TX522B 也会将数据块发回给主机。

在设计主机程序时，如果对时间要求比较严格，给 TX522B 发送数据或命令结束后，在等待接收 TX522B 返回数据时，先检测 TX522B 发出的 STX 信号并立即给出应答 ACK，然后接收 TX522B 发回的数据块。这种方式可以最快的接收 TX522B 的数据，提高实时性。

由于主机不给 ACK 信号，在 TX522B 接收命令后发出返回数据启动信号 STX 20ms 后也会将数据块发出，因此主机也可以在向 TX522B 发送命令后，不做任何检测和应答而直接等待 20ms 后的应答。这对于使用串口调试助手来调试 TX522B 特别有用。

5. TX522B 的 C51 函数

TX522B 提供了 C51 函数库，使用 C51 单片机的客户可直接调用这些函数库中的函数，其它系列的微控制器可参考 C51 函数库来进行移植，移植过程只要进行少量的修改。

5.1 函数列表

表 13 TX522B 提供的串行通讯库函数列表

函数名称	命令值	输入参数(发送)	输出参数（接收）	功能描述
TX_Config	0x52	--	--	复位且配置模块
TX_ConfigAnt	0x53	AntMode	--	配置模块的天线驱动模式
TX_Close	0x3F	--	--	关闭模块
TX_Get_Info	0x4F	--	*Info	读取模块信息
TX_Load_Key	0x4C	KeyAB,Sector,*Key	--	改变存贮在模块内密匙区中的密钥
TX_Get_CardSnr	0x10	ReqCode,*TagType, *Sak,*SnrLen, *Snr	--	激活卡片并获取卡号
TX_Write_Auth	0x11	KeyAB, Key_Sector , Block, idata *Data	--	带验证的写操作
TX_Ul_Write	0x76	Block,*Data	--	向 UltraLight 卡中相应块写入 4 字节数据
TX_Read_Auth	0x12	KeyAB, KeySector , Block	*Data	带验证的读操作
TX_WriteValue_Auth	0x13	KeyAB,Key_Sector, Block, *Value	--	带验证的写值块操作
TX_Value_Auth	0x14	KeyAB,Key_Sector, ValueMode,Block, *Value, Trans_Block	--	带验证的值块操作
TX_ReadValue_Auth	0x15	KeyAB,Key_Sector, Block,	*Value	带验证的读值块操作
TX_Halt	0x45	--	--	将卡置于挂起模式
TX_Reset	0x4E	Msec	--	关闭天线输出数 ms，使卡复位
TX_Auto_Detect	0x80	TxMode, ReqCode, AuthMode, KEYAB, *Key, uchar Block		配置 TX522B 进入自动寻卡模式
TX_Read_ADDT	0x81	ReadMode	*ATQ *TxDrv, *SAK, *PiccSnr, *DTBfr	读取自动寻卡数据。主机也可以随时发送此命令取消自动寻卡
TX_Request	0x41	ReqCode	*TagType	请求卡，检查在有效范围内是否有卡
TX_Casc_Anticoll	0x74	Bcnt, Select_Code	*SNR	可实现三层防碰撞协议
TX_Casc_Select	0x75	Select_Code, *SNR	*Sak	可实现三层选择
TX_Auth2	0x72	KeyAB,Sector,Key_Sector	--	使用模块内部密钥区中的密码对指定的卡的扇区 Sector 进行验证
TX_Auth_Key	0x73	KeyAB,Sector,*Key	--	直接密码验证
TX_Write	0x47	Block ,*Data	--	向卡中指定块写入一 16 字节的数据块
TX_Read	0x46	Block	*Data	从卡中指定块中读出一个 16 字节的块
TX_Value	0x70	ValueMode,Block,*Value, Trans_Block	--	包含加、减、恢复函数，并可进行不同块之间的自动传送
TX_Set_Control_Bit	0x50	--	--	将控制位置为高电平
TX_Clr_Control_Bit	0x51	--	--	将控制位置为低电平
TX_Buzzer	0x60	Freguence,Opentm, Closetm,Repcnt	--	输出驱动蜂鸣器信号，能控制动作时间、间隙时间和重复次数

表 13 所有函数的返回值都是该函数执行后的状态结果，具体返回值请查看各函数说明。如果各函数有返回数据，则都是以指针的形式返回。

除表 13 提供的通过串口与 TX522B 通讯的库函数，表 14 列出了 5 个其它函数。

表 14 TX522B 提供的其它库函数列表

函数名称	输入参数(发送)	输出参数（接收）	功能描述
Delay_50us	num	--	延时 50us 的 num 倍
Serial_Init	Baud_Num	--	串口按指定波特率初始化

5.2 函数返回状态值列表

表 15 TX522 库函数的用到的状态值列表

名称	值	描述
OK, COMM_OK	0	函数调用成功
NO_TAG_ERR	1	在有效区域内没有卡
CRC_ERR	2	从卡中接收到了错误的 CRC 校验和
EMPTY	3	值溢出
AUTH_ERR	4	不能验证
PARITY_ERR	5	从卡中接收到了错误的校验位
CODE_ERR	6	通信错误
SENDR_ERR	8	在防冲突时读到了错误的串行码
KEY_ERR	9	证实密码错
NOT_AUTH_ERR	10	卡没有验证
BIT_COUNT_ERR	11	从卡中接收到了错误数量的位
BYTE_COUNT_ERR	12	从卡中接收了错误数量的字节
TRANS_ERR	14	调用 Transfer 函数出错
WRITE_ERR	15	调用 Write 函数出错
INCR_ERR	16	调用 Increment 函数出错
DECR_ERR	17	调用 Decrment 函数出错
READ_ERR	18	调用 Read 函数出错
COLL_ERR	24	冲突错
ACCESS_TIMEOUT	27	访问超时
QUIT	30	上一次了送命令时被打断
CHK_WR_OK	0	Check Write 正确
CHK_WR_FAILED	1	Check Write 出错
CHK_WR_COMP_ERR	2	Check Write:写出错（比较出错）
COMM_ERR	255	串行通信错误
MI_WRONG_VALUE	123	值块格式错误

5.3 函数描述

下面是 C51 函数声明，包含在头文件“TX_B.h”中，写应用程序时，将其包含在应用函数中即可。对于 C51 函数的调用，用户可不必关心数据块格式，只要理解函数的功能，输入、输出参数即可。当用户自己编写函数（包括非 C51 下的函数）时，就需了解数据块格式，必须按照数据块的格式来编写函数。

对于只需要读取卡片某个固定块数据的应用，用户只要了解和使用 TX_Load_Key 、TX_Auto_Detect 和 TX_Read_ADDT 即可。

对应大多数需要改写卡片数据的应用，用户一般只需要了解和使用 TX_Config、TX_Close、TX_Load_Key、TX_Get_CardSnr、TX_Write_Auth、TX_Ul_Write、TX_Read_Auth、TX_WriteValue_Auth、TX_Value_Auth、TX_ReadValue_Auth、TX_Halt、TX_Reset 即可。

5.3.1 配置—Config

函数原型：uchar TX_Config(void);

输入参数：无

输出参数：无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR。

功能描述: 对模块进行初始化，初始化成功后，模块上的指示灯将点亮，此时天线发射载波信号，任何进入天线感应区的卡可得电进入 IDLE 状态，可使用任一函数对卡进行操作。此时读卡芯片完全被激活，所消耗的电流最大。模块上电后，模块内部会自动执行初始化，指示灯将点亮，因此上电后，用户可不用执行该函数，而直接进行其它操作。该函数可用于不上电情况下的重新初始化。

注意：对模块的配置需要大约 2.5ms 的时间，因此调用该函数后需要等待 2.5ms 以上才能进行其他操作。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND: 0x52

LENGTH: 0

DATA[…]: 无

例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x52	0x00	none	0xAD	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK，QUIT，COMM_ERR 中的某一个

LENGTH: 0

DATA[…]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.2 配置—ConfigAnt

函数原型：uchar TX_ConfigAnt(void);

输入参数：TxMode:天线驱动模式，0x00= TX1 和 TX2 关闭；0x01=TX1 驱动；0x02 = TX2 驱动；0x03 = TX1 和 TX2 同时驱动

输出参数：无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR。

功能描述: 对模块天线驱动模式进行配置。模块默认为 TX1 和 TX2 同时驱动。只要要进行双天线操作时，才会使用到该函数。

分体式模块不支持该函数。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR:　　0 （可自定义）

COMMAND:　0x53

LENGTH:　　1

DATA[0]:　　TxMode

例如：配置为 TX1 驱动的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x53	0x01	0x01	0xAC	0x03

TX522→主机（响应模式）：

SEQNR:　　0

STATUS:　　OK，QUIT，COMM_ERR 中的某一个

LENGTH:　　0

DATA[…]:　　无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.3 关闭－Close

函数原型：uchar TX_Close(void);

输入参数：无

输出参数：无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，　COMM_ERR。

功能描述: 此函数将关闭 TX522B 模块，指示灯熄灭，天线不发送载波信号，模块消耗的电流最小，在此状态在，模块不能使用。若要重新使用模块，需要调用 TX_Config()函数对 TX522B 重新进行配置。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR:　　0 （可自定义）

COMMAND:　0x3F

LENGTH:　　0

DATA[…]:　　无

例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x3F	0x00	none	0xC0	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK， COMM_ERR 中的某一个

LENGTH: 0

DATA[…]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.4 获取信息—Get Info

函数原型：uchar TX_Get_Info(uchar idata *Info);

输入参数：无

输出参数：模块的信息*Info，Info 为保存信息空间的首地址。Info[0]～Info[4]为模块类型标识，依次为‘5’，‘2’，‘2’，‘B’，0，Info[5]～Info[8]为模块的唯一序列号，Info[9]为固件版本号，高四位为版本号的整数，取值从 1 到 15，低四位为版本号的小数，取值从 0 到 9。

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR。

功能描述：获取 TX522B 模块的信息。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND: 0x4F

LENGTH: 0

DATA[…]: 无

例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x4F	0x00	none	0xB0	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK，QUIT，COMM_ERR 中的某一个

LENGTH: 10

DATA[0]：产品类型标识 0，‘5’

DATA[1]：产品类型标识 1，‘2’

DATA[2]：产品类型标识 2，’2’

DATA[3]：产品类型标识 3，’B’

DATA[4]: 产品类型标识 4， 0

DATA[5]：读卡芯片序列号 0

DATA[6]：读卡芯片序列号 1

DATA[7]：读卡芯片序列号 2

DATA[8]：读卡芯片序列号 3

DATA[9]：产品软件版本号

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x0a	9 字节产品型号及版本	0xXX	0x03

5.3.5 装载密钥—Load_Key

函数原型：uchar TX_Load_Key(uchar KeyAB,uchar Key_Sector,uchar idata *Key)

输入参数：KeyAB：密钥类型（1 字节）。可取值为 KeyAB＝0x00（KEYA）—密钥 A，或 KeyAB＝0x04（KEYB）—密钥 B。

Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15

*Key：需要装载到模块内密钥区的密钥（6 字节）

输出参数：无

函数返回： TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，AUTH_ERR，COMM_ERR。

功能描述：此函数的作用是将指定的密码（*Key）装载到模块内指定的密钥区（Key_Sector），并非改变 Mifare1 卡内扇区的密码。本函数只对模块进行操作，模块与卡之间没有数据传输。

模块内有 16 个密码区（区号 0——15），称它为密钥区号 Key_Sector。每个区分密钥 A（0x60）和密钥 B（0x61）两个，总共 32 个密码。装载成功后，可用该密钥对 Mifare1 卡进行验证。

在 M1 卡中也有 16 个存储区，称它为扇区号 Sector。若要改变 Mifare1 卡内的密钥，可在用原密码验证通过后，直接用写块数据 TX_Write()函数，将密码块改写。Mifare 卡出厂后的初始密钥为 6 个 FFH，A 和 B 密钥都一样。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND：0x4C

LENGTH: 8

DATA[0]: KeyAB

DATA[1]: Key_Sector

DATA[2]: Key[0]

…

DATA[7]: Key[5]

例如：往密匙 0 区装载密匙 A：0xff 0xff 0xff 0xff 0xff 0xff 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x4c	0x08	0x00 0x00 0xff 0xff 0xff 0xff 0xff 0xff	0xbb	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK，QUIT，AUTH_ERR，COMM_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.6 激活卡片并获取卡号—Get_CardSnr

函数原型：uchar TX_Get_CardSnr(uchar ReqCode, uchar idata *TagType, uchar idata *Sak, uchar idata *SnrLen, uchar idata *Snr);

输入参数： ReqCode：请求模式 ReqCode 取值为 1 或 0
ReqCode＝0（IDLE），请求天线范围内 IDLE 状态的卡（HALT 状态的除外）
ReqCode＝1（ALL），请求天线范围内的所有卡。

输出参数：
(1) *TagType：请求应答：2 个字节的卡片类型，其意义见 5.3.17 请求—Request。
(2) *Sak：最后一级选择应答的应答,其意义见 5.3.18 带层级设置的防碰撞—Casc_Anticoll。
(3) *SnrLen：返回卡片序列号的长度。
(4) *Snr：返回卡片的序列号。

函数返回：执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR 中的某一个。

功能描述：该命令为后面将要介绍的请求、防碰撞和选择三条命令的组合。成功执行该命令后即可进行验证及后续操作。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR: 0 （可自定义）
COMMAND：0x10
LENGTH: 1
DATA[0]: ReqCode
例如：以 IDLE 方式激活卡的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x10	0x01	0x00	0xee	0x03

TX522→主机（响应模式）：
SEQNR: 0
STATUS: OK，QUIT，COMM_ERR 中的某一个
LENGTH: 4 字节+序列号的长度，Mifare1 S50、S70、Light 卡：8 字节，
Mifare0 UltraLight 和 Mifare3 Desfire 卡：11 字节
DATA[0..1]: *TagType：请求应答，2 个字节的卡片类型
DATA[2]: *Sak：最后一级选择应答的应答
DATA[3]: *SnrLen：返回卡片序列号的长度
DATA[4..4+ SnrLen]: *Snr：返回卡片的序列号
例如：一张序列号为 0x007e0a42 的 Mifare1 S50 卡返回的数据

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x08	0x04 0x00 0x08 0x04 0x42 0x0A 0x7E 0x00	0xXX	0x03

例如：一张序列号为 0x000000007e0a42 的 Mifare UltraLight 卡返回的数据

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x0b	0x44 0x00 0x00 0x07 0x42 0x0A 0x7E 0x00 0x00 0x00 0x00	0xXX	0x03

5.3.7 带验证的写—Write_Auth

函数原型：uchar TX_Write_Auth(uchar KeyAB, uchar Key_Sector ,uchar Block, uchar idata *Data)

输入参数：

(1) KEYAB--密钥 AB（1 字节）： 0x00——密钥 A
0x04——密钥 B

(2) Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15。

(3) Block--卡块号（1 字节）： S50：1~63
S70：1~255

(4) *Data：16 字节数据指针，Data 为写入的 16 字节数据的首地址。

输出参数： 无

函数返回：执行后可能返回：OK，QUIT，NO_TAG_ERR，CRC_ERR，NOT_AUTH_ERR，PARITY_ERR ， BIT_COUNT_ERR ， COMM_ERR ， CHK_WR_FAILED ， CHK_WR_COMP_ERR 中的某一个。

功能描述：该函数在 Get_CardSnr 后执行，先对卡内某一块进行验证，成功后对指定块进行写操作（只要访问条件允许），其中包括位于扇区尾的密码块，这是更改密码的唯一方法。该函数在写入数据后会立即进行读操作并进行数据写入正确性判断。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR: 0 （可自定义）
COMMAND: 0x11
LENGTH: 19
DATA[0]: KEYAB
DATA[1]: Key_Sector
DATA[2]: Block
DATA[3]: 所要写的第一个字节
:
DATA[18]: 所要写的最后一个字节

例如：使用模块内部密匙 0 区的密匙 A 进行验证，往块 2 写入数据的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x11	0x13	0x00 0x00 0x02 16 字节数据	0xXX	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK，QUIT，NO_TAG_ERR，NOT_AUTH_ERR，WRITE_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

补充说明：

注意：在将数据写到卡片上的某一扇区时，一定要小心。因为有些 block 中存储了密码数据以及存储允许使能数据。特别是每一个扇区的 Block3 中存放了该扇区的存取条件，包含有 KEYA，KEYB 及该扇区的控制字。Mifare 1 卡片出厂时的 Block3 有缺省值，为：“a0a1a2a3a4a5ff078069b0b1b2b3b4b5”,共 16 个 Bytes。

涉及 Mifare 1 卡片的存储结构等信息，请参考 Mifare 1 卡片数据手册。

程序员在使用 Mifare 1 卡片做应用时，一定要清清楚楚记住每一个扇区的 Block3 的数据，这样也就记住了扇区的密码和存取控制字。否则，扇区的存储空间将不执行 Read/Write 等操作而失效。

任何试图用任何方式来读写不知密码的卡片或某一扇区都是徒劳无益的。

卡片应放在安全的地方，即不要放在离模块天线较近的地方。因为当模块对其它卡片执行某些指令时，有可能无意间对这一卡片进行了读/写等操作，从而操作卡片的失效。

5.3.8 写 UltraLight—Ul_Write

函数原型：uchar TX_Ul_Write(uchar Block, uchar *Data);

输入参数：Block：卡块号（1 字节）： 0～15

*Data: 4 字节数据指针，Data 为写入的 4 字节数据的首地址。

输出参数：无

函数返回：执行后可能返回：OK，QUIT，NO_TAG_ERR，CRC_ERR，NOT_AUTH_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR，CHK_WR_FAILED，CHK_WR_COMP_ERR 中的某一个。

功能描述：对 UltraLight 卡写入一个 4 字节的数据。此命令只对 UltraLight 有效。对 UltraLight 进行读操作与 mifare1 一样。该函数在写入数据后会立即进行读操作并进行数据写入正确性判断。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND: 0x47

LENGTH: 5

DATA[0]: Block

DATA[1]: 所要写的第一个字节

:

DATA[4]: 所要写的最后一个字节

例如：往 Ultralight 卡块 1 写入数据的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x47	0x05	0x01 4 字节数据	0xXX	0x03

TX522→主机（响应模式）：

SEQNR:　　0

STATUS:　　OK， QUIT， NO_TAG_ERR， NOT_AUTH_ERR， WRITE_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个

LENGTH:　　0

DATA[0]:　　无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.9 带验证的读—Read_Auth

函数原型：uchar TX_Read_Auth(uchar KeyAB, uchar Key_Sector ,uchar Block,uchar idata *Data)

输入参数：

(1) KEYAB--密钥 AB（1 字节）：　　0x00——密钥 A
　　0x04——密钥 B

(2) Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15。

(3) Block--卡块号（1 字节）：　　S50：1~63
　　S70：1~255

输出参数：*Data：Data 为读回 16 字节数据的首地址。

函数返回：执行后可能返回：OK，QUIT，NO_TAG_ERR，CRC_ERR，NOT_AUTH_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个。

功能描述：该函数在 Get_CardSnr 后执行，先对卡内某一块进行验证，成功后读 Mifare 卡中相应块的数据。密码数据不能被读取。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR:　　0 （可自定义）

COMMAND:　0x12

LENGTH:　　3

DATA[0]:　　KEYAB

DATA[1]:　　Key_Sector

DATA[2]:　　Block

例如：使用模块内部密匙 0 区的密匙 A 进行验证，读取块 2 的数据的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x12	0x03	0x00，0x00，0x02	0xEC	0x03

TX522→主机（响应模式）：

SEQNR:　　0

STATUS： OK， QUIT， NO_TAG_ERR， CRC_ERR， NOT_AUTH_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个

LENGTH： 16

DATA[0]： 所访问块的第一个字节

：

DATA[15]： 所访问块的最后一个字节

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x10	16 字节数据	0xXX	0x03

5.3.10 带验证的写值块操作—WriteValue_Auth

函数原型：uchar TX_WriteValue_Auth(uchar KeyAB, uchar Key_Sector , uchar Block, uchar idata *Value);

输入参数：

(1) KEYAB--密钥 AB（1 字节）： 0x00——密钥 A
0x04——密钥 B

(2) Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15。

(3) Block--卡块号（1 字节）： S50：1~63
S70：1~255

(4) *Value：4 字节数据指针，用来存储减少值或增加值，当进行恢复操作时，该值为空值。Value 是减少值或增加值存放的首地址，存放时，低地址存放高字节。

输出参数： 无

函数返回：执行后可能返回：OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个。

功能描述：该函数在 Get_CardSnr 后执行，先对卡内某一块进行验证，成功后往 Mifare 卡中相应块写入值块格式的数据（只要访问条件允许）。该函数在写入数据后会立即进行读操作并进行数据写入正确性判断。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR： 0 （可自定义）

COMMAND：0x13

LENGTH： 7

DATA[0]： KEYAB

DATA[1]： Key_Sector

DATA[2]： Block

DATA[3]： Value(LL)

DATA[4]： Value(LH)

DATA[5]： Value(HL)

DATA[6]： Value(HH)

例如：使用模块内部密匙 0 区的密匙 A 进行验证，将块 1 写入值块 0x00 0x00 0x55 0x11(低位)，相当于数值 21777 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x13	0x07	0x00 0x00 0x01 0x11 0x55 0x00 0x00	0xAE	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.11 带验证的值块操作—Value_Auth

函数原型：uchar TX_Value_Auth (uchar KeyAB, uchar Key_Sector ,uchar ValueMode, uchar Block, uchar idata *Value, uchar Trans_Block);

输入参数：

(1) KEYAB--密钥 AB（1 字节）： 0x00——密钥 A
0x04——密钥 B

(2) Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15。

(3) ValueMode：0xC0—减； 0xC1—加； 0xC2—恢复

(4) Block--卡块号（1 字节）： S50：1~63
S70：1~255

(5) *Value：4 字节数据指针，用来存储减少值或增加值，当进行恢复操作时，该值为空值。Value 是减少值或增加值存放的首地址，存放时，低地址存放高字节。

(6) Trans_Block：传输块地址，取值范围：S50：1～63；S70：1～255。

输出参数： 无

函数返回：执行后可能返回：OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个。

功能描述：该函数在 Get_CardSnr 后执行，先对卡内某一块进行验证，成功后对卡内的某一块进行加、减或数据备份，该块必须为值块格式，并支持自动传送。若卡块号与传输块号相同，则将操作后的结果写入原来的块内；若卡块号与传输块号不相同，则将操作后的结果写入传输块内，结果传输块内的数据被覆盖，原块内的值不变。当模式为“恢复”时，“值”无意义。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND: 0x14

LENGTH: 9

DATA[0]: KEYAB

DATA[1]: Key_Sector

DATA[2]: ValueMode

DATA[3]:　　Block
DATA[4]:　　Value(LL)
DATA[5]:　　Value(LH)
DATA[6]:　　Value(HL)
DATA[7]:　　Value(HH)
DATA[8]:　　Trans_Block

例如：使用模块内部密匙 0 区的密匙 A 进行验证，将块 1 的值减 3，然后传送到块 1 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x14	0x09	0x00 0x00 0xc0 0x01 0x03 0x00 0x00 0x00 0x01	0x21	0x03

TX522→主机（响应模式）：
SEQNR:　　0
STATUS:　OK,　QUIT,　NO_TAG_ERR,　BIT_COUNT_ERR,　TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个
LENGTH:　　0
DATA[0]:　　无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.12 带验证的读值块操作—ReadValue_Auth

函数原型：uchar TX_ReadValue_Auth(uchar KeyAB, uchar Key_Sector , uchar Block, uchar idata *Value);

输入参数：

(1)　KEYAB--密钥 AB（1 字节）：　　0x00——密钥 A
　　　　　　　　　　　　　　　　0x04——密钥 B

(2)　Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15。

(3)　Block--卡块号（1 字节）：　　S50：1~63
　　　　　　　　　　　　　　　　S70：1~255

输出参数：　*Value：Value 为读回 4 字节数据的首地址。

函数返回：执行后可能返回：OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个。

功能描述：该函数在 Get_CardSnr 后执行，先对卡内某一块进行验证，然后读出值块。注意读出值块时，会对值块的正确性进行判断，如果值块不合法，返回错误，不会返回值块数据。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR:　　0 （可自定义）
COMMAND：0x15
LENGTH:　　3
DATA[0]:　　KEYAB

DATA[1]: Key_Sector
DATA[2]: Block
例如：使用模块内部密匙 0 区的密匙 A 进行验证，读块 1 的值块的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x15	0x03	0x00 0x00 0x01	0xE8	0x03

TX522→主机（响应模式）：
SEQNR: 0
STATUS: OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个
LENGTH: 4
DATA[0]: Value(LL)
DATA[1]: Value(LH)
DATA[2]: Value(HL)
DATA[3]: Value(HH)
例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x04	0x0E 0x55 0x00 0x00	0xAE	0x03

5.3.13 暂停—Halt

函数原型：uchar TX_Halt(void)
输入参数：无
输出参数：无
函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR。

功能描述：将天线区所选择卡置为挂起状态。如果要进行重新选择，则应用 ALL 模式调用 TX_Request 命令。如果要进行重新选择,也可以将卡离开天线操作区再进入，或执行复位函数 TX_Reset()。

可以配合使用 TX_Request()和 TX_Halt()函数，进行一次性扣费，如卡进入感应区后只扣一次（一元钱），离开后，下次进入再扣一次，若卡在感应区内停留时间较长，也不会扣多一次。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR: 0 （可自定义）
COMMAND：0x45
LENGTH: 0
DATA[0]: 无
例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x45	0x00	none	0xba	0x03

TX522→主机（响应模式）：

SEQNR:　　0
STATUS:　　OK，QUIT，COMM_ERR 中的某一个
LENGTH:　　0
DATA[0]:　　无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.14 复位—Reset

函数原型：uchar TX_Reset(uchar Msec);

输入参数：Msec：取值 0~255，模块上射频电路关闭时间（以 ms 毫秒为单位），Msec＝0 时，一直关闭。

输出参数：　无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR 中的某一个。

功能描述：该函数使模块上的射频电路关闭，关闭的时间由参数 Msec 指定，若 Msec=0，射频电路将一直处于关闭状态，一直到下一个 TX_Request 命令到来。关闭射频电路能使天线内的所有卡复位。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR:　　0 （可自定义）
COMMAND：0x4E
LENGTH:　　1
DATA[0]:　　Msec
例如：将天线信号关闭 1ms 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x4e	0x01	0x01	0xb0	0x03

TX522→主机（响应模式）：
SEQNR:　　0
STATUS:　　OK，QUIT，COMM_ERR 中的某一个
LENGTH:　　0
DATA[0]:　　无
例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.15 自动寻卡—Auto_Detect

函数原型：uchar TX_Auto_Detect(uchar ADMode, uchar TxMode, uchar ReqCode,
uchar AuthMode, uchar KEYAB,
uchar idata *Key, uchar Block)

输入参数：

(1) ADMode--自动检测模式（1 字节）

B7	B6	B5	B4	B3	B2	B1	B0
RFU 00000				1：最后执行 Halt() 命令；0：无动作。	数据输出后 1：继续检测；0：不继续检测。	RFU = 1	当 UART 接口，检测到有卡时：0：串口不发送；1：串口主动回发，发送的数据格式见说明 B。

将自动寻卡方式配置为寻到卡后串口主动回发数据，然后主机使用 UART 接收中断，主机程序中采用 UART 接收中断接收模块返回的数据。通过这种方式，用户可以不使用 INT_OUT 管脚。

使用 UART 接收中断而不使用中断管脚时，用户程序中的关中断时间不能大于 20ms，因为 TX522B 模块在寻到卡后，主动向主机发送请求 STX，如果在 20ms 内得不到应答就直接返回数据。如果主机程序中关中断的时间超过 20ms，则推荐使用 INT_OUT 管脚，INT_OUT 管脚在自动寻卡成功后会变为低电平，直到自动寻卡数据被主机读走才变为高电平。

(2) TxMode--天线驱动方式（1 字节），0x00= TX1 和 TX2 关闭；0x01=TX1 驱动；0x02 = TX2 驱动；0x03 = TX1 和 TX2 同时驱动。

(3) ReqCode--请求代码（1 字节）： 0x00——IDLE
0x01——ALL

(4) AuthMode--验证命令（1 字节）： ‘E’——用 E2 密码验证
‘F’——用直接密码验证
0——不验证

(5) KEYAB--密钥 AB（1 字节）： 0x00——密钥 A
0x04——密钥 B

(6) Key--密钥： 若验证命令=‘E’，则为密钥区号（1 字节）
若验证命令=‘F’，则为密钥（6 字节）

(7) Block--卡块号（1 字节）： S50：0~63
S70：0~255

输出参数：无。

返回值：TX522B 执行命令后的状态，可能的状态值如下：COMM_ERR、COMM_OK 等。

功能描述：用于配置 TX522B 进入自动寻卡模式。配置好后模块不断去读取卡片，检测是否有卡片进入天线区域。如果有卡片进入，则自动读取卡号，并根据配置验证卡片读取数据，并根据配置产生中断或者主动回发卡号和数据。而回发完数据之后，再根据配置继续进行自动寻卡，或者退出自动寻卡模式。注意回发数据时，如果卡片不拿开,则只有第一次检测到时才回发数据，但卡片指示 INT_OUT 一直有效。

自动寻卡命令会保存到内部 E2ROM 中，掉电保存。下一次上电或者复位后继续有效。

数据块格式描述：

1. 主机→TX522 （命令模式）：

SEQNR: 0 （可自定义）
COMMAND: 0x80
LENGTH: 若验证命令='E'，则为 7
若验证命令='F'，则为 12
若验证命令= NULL，则为 4
DATA：

验证命令	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]~[10]	D[11]
NULL	ADMode	TxMode	ReqCode	AuthMode	-	-	-	-	-
'E'					KEYAB	密钥扇区号	块号	-	-
'F'					KEYAB	密钥			块号

例如：当检测到有卡时产生中断，串口主动输出，以 IDLE 方式激活卡，双天线驱动，用直接密码验证密钥 A（密码为 0xff 0xff 0xff 0xff 0xff 0xff），读出第 1 块数据，数据输出完毕后停止自动寻卡。则数据帧如下：

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x80	0x0C	0x03 0x03 0x00 0x46 0x00 0xff 0xff 0xff 0xff 0xff 0xff 0x01	0xXX	0x03

2. TX522→主机（响应模式）：
SEQNR: 0
STATUS: OK 等
LENGTH: 0
DATA[0]: 无
例如，数据帧：

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

3. 说明
A. 从机应答说明模块已经正确地收到此命令。此命令相当以下命令的组合：
请求——防碰撞——选择——验证（若 AuthMode！=0）——读取（若 AuthMode！=0）——挂起（若 ADMode.3=1）
B. 在自动检测模式使能后，若允许串口主动发送（即 ADMode.0=1），则检测到卡片后，发送的数据信息格式如下：

```
uchar TX_ADRcv(uchar idata *TxDrv,uchar idata *ATQ,uchar idata *SAK,uchar idata *PiccSnr,uchar idata *DTBfr)
```

TX522→主机 主动回发数据帧格式如下：
SEQNR: 0
STATUS: OK 等
LENGTH: 若验证命令不为 NULL，则为：21+序列号长度（S50/S70 卡应该为 25）
若验证命令为 NULL，则为：5+序列号长度（S50/S70 卡应该为 9）
DATA：

D[0]	D[1]~ [2]	D[3]	D[4]	D[5]~[8]	D[9]~[24]
TxDrv (天线驱动)	ATQ (请求应答)	SAK (选择应答)	SnrLen (卡号长度)	PiccSnr (卡号)	Data （数据，如果验证命令不为 NULL）

其中，TxDrv 的含义如下：

TxDrv 天线驱动

B7	B6	B5	B4	B3	B2	B1	B0
RFU 000000						00：TX1、TX2 关闭 01： TX1 驱动 10： TX2 驱动 11：TX1、TX2 同时驱动	

例如：检测到序列号为 0x007e0a42 的 S50 卡，从机将发送的数据为：

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x09	0x03 0x04 0x00 0x08 0x04 0x42 0x0a 0x7e 0x00	0xXX	0x03

串口主动发送之后，模块状态由 *ADMode.2* 位来决定。若 *ADMode.2*=1，则继续进入自动检测模式；否则结束自动检测模式，主机可以发送其它任何命令。

C. 若 *ADMode.1*=1，则模块检测到卡后产生中断信号。产生中断后，主机可以通过两种方式读回数据：

- 一是直接读取，可通过以下函数实现：

```
uchar TX_ADRcv(uchar idata *TxDrv,uchar idata *ATQ,uchar idata *SAK,uchar idata *PiccSnr,uchar idata *DTBfr)
```

这样读取之后的模块状态由 *ADMode.2* 位来决定。若 *ADMode.2*=1，则继续进入自动检测模式；否则结束自动检测模式，主机可以发送其它任何命令。

- 二是通过读取自动检测数据命令（Cmd = 0x81）读取。

在任何通信方式下，通过读自动检测数据命令读取数据之后的模块状态由该函数的参数来决定，参见此命令的说明。

D. 在自动检测模式期间，主机可以随时发出读取自动检测数据命令，读取自动检测数据、查询自动检测状态、取消或继续自动检测。

E. 验证和读命令只对 Mifare1 卡有效。

F. 在自动检测模式期间，若主机发送任何除读自动检测数据外的，且数据长度小于 3（帧长小于 8）的命令，将退出自动检测模式。如请求 *PiccRequest()*命令。在此期间，模块将不接收数据长度大于 2（帧长大于 7）的命令。

5.3.16 读取自动检测数据—Read_ADDT

函数原型：uchar TX_Read_ADDT(uchar ReadMode,uchar idata *TxDrv, uchar idata *ATQ, uchar idata *SAK,uchar idata *PiccSnr,uchar idata *DTBfr)

输入参数：ReadMode – 读模式

B7	B6	B5	B4	B3	B2	B1	B0
RFU 0000000							数据读回之后： 1：继续检测； 0：取消检测。

输出参数： TxDrv – 天线驱动方式（1 字节）；
ATQ – 请求应答（2 字节）；
SAK – 选择应答（1 字节）；
PiccSnr – 卡号（N 字节）；
DTBfr –数据（16 字节，如果验证命令不为 NULL）；

返回值：OK NO_TAG_ERR 等

功能描述：当模块设置为非主动回发数据模式，则当接收到中断后，主机可以用此函数读取自动寻卡数据。主机也可以随时发送此命令取消自动寻卡。

数据块格式描述：

1. 主机→TX522 （命令模式）：

SEQNR: 0 （可自定义）

COMMAND：0x81

LENGTH： 1

DATA[0]: ReadMode

例如，读取自动检测数据之后取消自动检测，则数据帧如下：

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x81	0x01	0x00	0x7f	0x03

2. TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK—成功；NO_TAG_ERR – 未检测到卡片；-100 – 该命令未执行。

LENGTH： 若验证命令不为 NULL，则为：21+序列号长度
若验证命令为 NULL，则为：5+序列号长度

DATA：

验证命令	D[0]	D[1]~[2]	D[3]	D[4]	D[5]~[8]	D[9]~[24]
NULL	TxDrv (天线驱动)	ATQ（请求应答）	SAK（选择应答）	序列号长度	序列号	序列号（根据序列号长度而定）
‘E’、’F’					序列号	扇区数据

例如：检测到序列号为 0x007e0a42 的 S50 卡，从机发送的数据为：

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x09	0x03 0x04 0x00 0x08 0x04 0x42 0x0a 0x7e 0x00	0xXX	0x03

5.3.17 请求—Request

函数原型：uchar TX_Request(uchar ReqCode,uchar idata *TagType);

输入参数： ReqCode：请求模式 ReqCode 取值为 1 或 0
ReqCode＝0（IDLE），请求天线范围内 IDLE 状态的卡（HALT 状态的除外）
ReqCode＝1（ALL），请求天线范围内的所有卡。

输出参数：*TagType：2 个字节的卡片类型，*TagType 低字节 *(TagType+1)高字节，对于 mifare 1 卡，返回类型为 0x0004,则*TagType ＝ 0x04 ，*(TagType+1)＝0x00。 当发生错误时，不返回任何内容（LENGTH=0）。

*TagType 的含义如表 16 所示：

表 16 *TagType 的含义

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
保留								序列号大小 00:4bytes 01:7bytes 10:10bytes		保留	任何位为 1 时，则为比特帧防碰撞方式				

各种常用卡返回的数据如下表 17 所示：

表 17 常用卡的*TagType 值

卡类型	*TagType	卡类型	*TagType
Mifare1 S50	0x0004	SHC1101	0x0004
Mifare1 S70	0x0002	SHC1102	0x3300
Mifare Light	0x0010	11RF32	0x0004
Mifare UltraLight	0x0044		

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，NO_TAG_ERR，BIT_COUNT_ERR，COMM_ERR。

功能描述：检查在 TX522B 有效范围内是否有卡存在。卡片进入天线区域后，能获能能量，从而得电复位处于 IDLE 模块，TX_Request 函数可用 ALL 或 IDLE 任意一种模式进行请求，卡片均能响应，并返回卡片类型号 TagType（2 个字节）。在选择一张新的卡时必须调用该函数。若对某一张卡成功进行了挂起操作（TX_Halt 命令），卡片将进入 Halt 状态，此时的卡只能响应 ALL 模式的请求，除非该卡离开天线区域然后再重新进入。

给 TX522B 发送 0x41 命令完成请求卡。注意：对同一张卡（不进入 HALT 状态）连续进行请求时，总是一次成功一次失败。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR： 0 （可自定义）

COMMAND： 0x41

LENGTH： 1

DATA[0]： ReqCode

例如：请求天线范围内所有的卡的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x41	0x01	0x01	0xBE	0x03

TX522→主机（响应模式）：

SEQNR： 0

STATUS： OK，QUIT，NO_TAG_ERR，BIT_COUNT_ERR，COMM_ERR 中某一个

LENGTH： 2 (当发生错误时，不返回任何内容（LENGTH=0）)

DATA[0]： tagtype (低字节)

DATA[1]： tagtype (高字节)

例如：S50 卡返回的数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x02	0x04 0x00	0xF9	0x03

补充说明：

TX_Request (IDLE)的使用是很重要的，它可以防止 TX522 多次选择同一卡片。当某一张卡片在天线的有效的工作范围（距离）内，TX_Request (IDLE)指令在成功地读取这一张卡片之后，将一直等待卡片的使用者拿走这一张卡片，直到有新一张的卡片进入天线的有效的工作范围（距离）内。当然，这里的“新一张的卡片”亦可以是刚刚拿开那张卡片。

TX_Request (IDLE)指令是非连续性的读卡指令，只读一次。但有个例外，当某一次

Request 指令读卡片失败时，例如，卡片没能通过密码认证或其他原因而出错时，TX_Request (IDLE)指令将连续地读卡，直到读卡成功才进入非连续性的读卡模式。

TX_Request (IDLE)指令适用于那些需要有人工干预的场合。

TX_Request (ALL)指令的使用和 TX_Request (IDLE)指令刚巧相反，TX_Request (ALL)指令是连续性的读卡指令。当某一张卡片在天线的有效的工作范围(距离)内，TX_Request (ALL)指令在成功地读取这一张卡片之后，进入对卡片的其他操作。如果其他操作完成之后，程序员执行 TX_Request (ALL)指令操作，则 TX_Request (ALL)指令将连续性地再次进行读卡操作，而不管这张卡片是否被拿走。只要有一张卡片进入天线的有效的工作范围（距离）内，TX_Request (ALL)指令将始终连续性地再次进行读卡操作。

TX_Request (ALL)指令适用于那些不需要有人工干预的场合，即全自动的场合。例如，宾馆，酒店，高级写字楼等场所的门禁控制系统（Door Access Control)，高速公路，停车场等的不停车收费系统（Non-Stop Road Tolling)，等等。

后面即将介绍的 TX_Halt()命令函数使被选中的卡进入 HALT 模式，进入该模式的卡只能用 ALL 方式进行请求，也即它对 IDLE 方式的请求不响应。若要使它再次响应 IDLE 请求，唯一方式是使卡复位，如执行 TX_Reset()函数，或使卡离开天线感应区后再进入。

可以配合使用 TX_Request()和 TX_Halt()函数，进行一次性扣费，如卡进入感应区后只扣一次（一元钱)，离开后，下次进入再扣一次，若卡在感应区内停留时间较长，不得扣多于一次。方法如下：

```
while(1)
{
    while(TX_Request(IDLE, databuf)!= OK);
    …
    扣费操作
    …
    TX_Halt();
}
```

5.3.18 带层级设置的防碰撞—Casc_Anticoll

函数原型：uchar TX_Casc_Anticoll(uchar Select_Code,uchar Bcnt,uchar idata *SNR);

输入参数：Select_Code：防碰撞层级编码：一层（ANTICOLL1）—0x93；二层（ANTICOLL2）—0x95；三层（ANTICOLL3）—0x97。Bcnt：预选卡已经知道的序列号的的位数，通常都设置 Bcnt=0。

输出参数：*SNR：4 个字节的卡的序列号，低字节放在低地址处*SNR 低字节 *(SNR+i)高字节。若卡的序列号超过 4 个字节（如 Mifare UltraLight)，序列号不完整，则最低字节的值为 0x88，表示需要进行更高一级的防碰撞 TX_Casc_Anticoll（0x95)。

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，NO_TAG_ERR，BIT_COUNT_ERR，COMM_ERR。

功能描述: 可选级数的防碰撞操作。该函数必须在调用 TX_Request 命令后立即调用。当多个卡片位于天线区域时，执行函数后，能得到序列号最大的那个卡片的序列号中的 4 个字节。卡的序列号长度有三种：4 字节、7 字节和 10 字节。4 字节的只要用一级防碰撞即可得到完整的序列号，如 Mifare1 S50 S70 等；7 字节的要进行二级防碰撞才能得到完整的序列号，前一级所得到的序列号的最低字节为级联标志 0x88，在序列号内只后 3 字节可用，后一级防碰撞能得到 4 字节序列号，两者按顺序连接即为 7 字节序列号，如 UltraLight 和

DesFire 等；10 字节的以此类推，但至今没有此类卡。需要进行二级防碰撞操作，可通过修改参数 Select_Code 来实现。第一级使用 Select_Code＝0x93，第二级使用 Select_Code＝0x95，对于现在所有得 Mifare 卡，不需要进行三级防碰撞，因为所有 Mifare 卡的序列号最大为 7 个字节。当知道了所要选择卡的序列号后，就没有必要执行该 TX_AntiColl 函数。此时，调用 TX_Request 后，直接调用 TX_Select 函数即可。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR： 0 （可自定义）

COMMAND：0x74

LENGTH： 2

DATA[0]： Select_Code

DATA[1]： Bcnt

例如：第一级防碰撞的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x74	0x02	0x93 0x00	0x1A	0x03

TX522→主机（响应模式）：

SEQNR： 0

STATUS：OK，QUIT，NO_TAG_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个

LENGTH：4

DATA[0]：卡的序列号第 1 字节（最低字节）

DATA[1]：卡的序列号第 2 字节

DATA[2]：卡的序列号第 3 字节

DATA[3]：卡的序列号第 4 字节

例如：返回序列号 0x007e0a42 的数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x04	0x42 0x0A 0x7E 0x00	0xXX	0x03

补充说明：

符合 ISO14443A 标准卡的序列号都是全球唯一的，正是由于唯一性，才能实现防碰撞的算法逻辑。如果有多张卡同时在天线感应区内则这个函数能够找到一张序列号较大的卡来操作。实际上由于天线辐射的磁场能量有限，同时在天线感应区内的所有卡都要从辐射场中吸收，因此同时在天线感应区内的卡不能太多，否则没有一张卡能获得足够的能量来正常工作。

在执行选择 TX_Select 命令之前，若已知所要操作的卡的序列号，则可跳过此步，直接执行选择 TX_Select 命令；若不知道卡的序列号，则必须调用防碰撞函数，得到感应区内卡的序列号。

程序员在发送 TX_Casc_Anticoll 指令，TX522 将返回 SNR，一般地应在程序中对所接收到的 SNR 进行校验，以确保数据的正确性。具体的方法是对所接收到的 SN 的 bit 位进行异或校验。这个工作已经在 TX522B 中完成，用户不必再进行校验。

对于 MIFARE 1 卡片来说，返回某一张卡片的有效序列号 SNR（4 个字节）可能为：007e0a42h。

5.3.19 带层级设置的选择—Casc_Select

函数原型：uchar TX_Casc_Select(uchar SelectCode, uchar idata *SNR,uchar idata *Sak);

输入参数：Select_Code：防碰撞层级编码：一层（ANTICOLL1）—0x93；二层（ANTICOLL2）—0x95；三层（ANTICOLL3）—0x97。该参数应该与之前的 TX_Casc_Anticoll()中的 Select_Code 参数相同。

*SNR：前一次防碰撞返回的卡的序号，或已知的卡的序列号。无符号 4 字节，低字节放在低地址处。如果卡得序号号大于 4 个字节，则*SNR 的最低字节值为 0x88，表示需要进行更高一级的防碰撞。

输出参数：*Sak：是否选择成功的应答,其意义如表 18 所示：

表 18 *Sak 含义

b7	b6	b5	b4	b3	b2	b1	b0
RFU			RFU		判断是否选择完成	RFU	

bit2 位用来判断是否还有序列号未读出，bit2＝0 时，表示所有序列号输入完毕，选择成功，bit2＝1 时表示序列号没有全部输入，选择没有完全成功，还要进行下一级的防碰撞和选择操作。

当*Sak＝xxxxx1xx 时，序列号没有完成，还要进行下一级的防碰撞和选择操作。

当*Sak＝xx1xx0xx 时，选择成功，该卡符合 ISO/IEC 14443-4 标准。

当*Sak＝xx0xx0xx 时，选择成功，该卡不符合 ISO/IEC 14443-4 标准。

各种常用卡返回的数据如下所表 19 所示：

表 19 常用卡的*Sak 的值

卡类型	*Sak	卡类型	*Sak
Mifare1 S50	0x08	Mifare0 UltraLight	0x04
Mifare1 S70	0x18	SHC1101	0x22
Mifare1 Light	0x01	11RF32	0x08

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK, QUIT, NO_TAG_ERR, CRC_ERR, PARITY_ERR, BIT_COUNT_ERR,COMM_ERR。

功能描述：该函数选择某一个序列号的卡，与之建立通信连接，并返回是否选择成功的应答*Sak。该函数一般与 TX_Casc_Anticoll()配对使用，进行多级防碰撞后卡的选择。参数 Select_Code 表示防碰撞的级数，应该与该函数执行前的 TX_Casc_Anticoll()中的 Select_Code 相同。在任意一个防碰撞函数成功执行后，或在任何时候当程序员想实际地与已知序列号的卡片进行通信时，必须使用 TX_Casc_Select 或后面得 TX_Select 函数，以建立与所选卡的通信。卡的序列号长度有三种：4 字节、7 字节和 10 字节。4 字节的只要用一级选择即可,如 Mifare1 S50 S70 等；7 字节的要用二级选择才能完成，如 UltraLight 和 DesFire 等。如果需要进行二级选择，第一级选择输入的序列号为，第一级防碰撞所得到的序列号,其中最低字节为级联标志 0x88，只后 3 字节可用，第二级选择输入得序列号为第二级防碰撞得到 4 字节序列号。在程序中可用*Sak 的 bit2 位来判断是否还有序列号未读出，如 if(*SAK & 0x04){…}。

此外，当模块成功选择一张卡片后，模块内部会记录下该卡片的的序列号，并供其它后续需要卡号的函数使用，因此后续的操作函数（如 TX_Auth）都不需再输入卡片的序列号。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND：0x75
LENGTH：　5
DATA[0]：　Select_Code
DATA[1]：　卡的序列号第 1 字节（最低字节）
DATA[2]：　卡的序列号第 2 字节
DATA[3]：　卡的序列号第 3 字节
DATA[4]：　卡的序列号第 4 字节
例如：第一级选择，UID 为 0x007e0a42 数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x75	0x05	0x93 0x42 0x0A 0x7E 0x00	0xXX	0x03

TX522→主机（响应模式）：
SEQNR：　0
STATUS：　OK, QUIT, NO_TAG_ERR, CRC_ERR, PARITY_ERR, BIT_COUNT_ERR, COMM_ERR 中的某一个
LENGTH：　1
DATA[0]：　Sak
例如：选择 S50 卡的数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x01	0x08	0xf6	0x03

5.3.20 证实 2—Auth2

函数原型：uchar TX_Auth2(uchar KeyAB, uchar Sector, uchar Key_Sector);

输入参数：KeyAB：密钥类型（1 字节）。可取值为 KeyAB＝0x00（KEYA），利用密钥 A 进行验证，或 KeyAB＝0x04（KEYB），利用密钥 B 进行验证。

Sector：所要验证的卡扇区号（也即将要访问的卡的扇区号），取值范围 0～39，能用于 S70 卡。

Key_Sector：模块内的密钥区号（1 字节）：取值范围 0～15。

输出参数：　无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，NO_TAG_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR。

功能描述：使用模块内部密钥区中 Key_Sector 中的密码对指定的卡的扇区 Sector 进行验证，若卡 Sector 区中的密码与存储在模块内 Key_Sector 中的密码相同，则验证成功，返回 OK。

该函数也依赖与 TX_Load_Key 函数曾经成功执行过，因为模块内部密码区（Key_Sector）中的密码要由 TX_Load_Key 函数事先装载。该函数适用于对于所有卡来说密码相同的应用，密钥的装载可以在一个安全的场合一次性装入。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR：　0 （可自定义）

COMMAND：0x72
LENGTH：　3
DATA[0]：　KeyAB
DATA[1]：　Sector
DATA[2]：　Key_Sector
例如：用密匙 0 区的密匙 A 证实卡的扇区 0 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x72	0x03	0x00 0x00 0x00	0x8e	0x03

TX522→主机（响应模式）：
SEQNR：　0
STATUS：　OK， QUIT， NO_TAG_ERR， PARITY_ERR， BIT_COUNT_ERR，COMM_ERR 中的某一个
LENGTH：　0
DATA[0]：　无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.21 直接密码证实—Auth_Key

函数原型：uchar TX_Auth_Key (uchar KeyAB, uchar Sector, uchar idata *Key);

输入参数：KeyAB：密钥类型（1 字节）。可取值为 KeyAB＝0x00（KEYA），利用密钥 A 进行验证，或 KeyAB＝0x04（KEYB），利用密钥 B 进行验证。

Sector：所要验证的卡扇区号（也即将要访问的卡的扇区号），取值范围 0～39，能用于 S70 卡。

Key：用于证实的密码首地址，,应在外部定义一个共 6 个字节的数组用于存放密码

输出参数：　无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，NO_TAG_ERR，AUTH_ERR, PARITY_ERR，BIT_COUNT_ERR，COMM_ERR。

功能描述：　TX_Auth2 函数都必须要依赖于 TX_Load_Key()函数的曾经的成功执行，在进行证实之前，一定要确认正确的密钥已经存于模块的密钥区内。而 TX_Auth_Key 函数则将密钥存放于 Key 指针所指向的 6 个字节存储区内，TX_Auth_Key 函数执行时不对密钥区进行操作，因此也称为直接密码证实。若卡中的密钥与所传输的密码相匹配。则证实成功，函数将返回 OK。

直接密码证实一般用于对每一张卡来说密钥都不同的应用，如在使用安全模块（PSAM 卡）的消费应用中，消费机首先将卡的序列号读出，然后与 PSAM 卡中消费主密钥一起生成导出密钥，然后直接用导出密钥与卡的一个应用扇区相互证实。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR：　0 （可自定义）

COMMAND：0x73
LENGTH：　8
DATA[0]：　KeyAB
DATA[1]：　Sector
DATA[2]：　Key[0]
…
DATA[7]：　Key[5]
例如：用密码 0xff 0xff 0xff 0xff 0xff 0xff 证实扇区 0 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x73	0x08	0x00 0x00 0xff 0xff 0xff 0xff 0xff 0xff	0x84	0x03

TX522→主机（响应模式）：
SEQNR：　0
STATUS：　OK，QUIT，NO_TAG_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个
LENGTH：　0
DATA[0]：　无
例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.22 写—Write

函数原型：uchar TX_Write(uchar Block,uchar idata *Data)
输入参数：Block：卡块号（1 字节）：　S50：1～63；S70：1～255
*Data：16 字节数据指针，Data 为写入的 16 字节数据的首地址。
输出参数：　无
函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，NO_TAG_ERR，CRC_ERR，NOT_AUTH_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个。

功能描述：对卡内某一块进行验证成功后，即可对同一扇区的各个块进行写操作（只要访问条件允许），其中包括位于扇区尾的密码块，这是更改密码的唯一方法。

Mifare 卡中一个块的数据是 16 字节，因此读写一次均是 16 个字节。所读块号必须与之前所验证的块号在同一个扇区内，mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区。

数据块格式描述：
主机→TX522 命令模式）：
SEQNR：　0 （可自定义）
COMMAND：0x47
LENGTH：　17
DATA[0]：　Block
DATA[1]：　所要写的第一个字节
：
DATA[16]：　所要写的最后一个字节

例如：往块 2 写入数据的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x47	0x11	0x02 16 字节数据	0xXX	0x03

TX522→主机（响应模式）：

SEQNR:　　0

STATUS:　　OK， QUIT， NO_TAG_ERR， NOT_AUTH_ERR， WRITE_ERR， BIT_COUNT_ERR，COMM_ERR 中的某一个

LENGTH:　　0

DATA[0]:　　无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

补充说明：

注意：在将数据写到卡片上的某一扇区时，一定要小心。因为有些 block 中存储了密码数据以及存储允许使能数据。特别是每一个扇区的 Block3 中存放了该扇区的存取条件，包含有 KEYA，KEYB 及该扇区的控制字。Mifare 1 卡片出厂时的 Block3 有缺省值，为："a0a1a2a3a4a5ff078069b0b1b2b3b4b5",共 16 个 Bytes。

涉及 Mifare 1 卡片的存储结构等信息，请参考 Mifare 1 卡片数据手册。

程序员在使用 Mifare 1 卡片做应用时，一定要清清楚楚记住每一个扇区的 Block3 的数据，这样也就记住了扇区的密码和存取控制字。否则，扇区的存储空间将不执行 Read/Write 等操作而失效。

任何人试图用任何方式来读写不知密码的卡片或某一扇区都是徒劳无益的。

卡片应放在安全的地方，即不要放在离模块天线较近的地方。因为当模块对其它卡片执行某些指令时，有可能无意间对这一卡片进行了读/写等操作，从而操作卡片的失效。

5.3.23 读—Read

函数原型：uchar TX_Read(uchar Block,uchar idata *Data)

输入参数：Block：卡块号（1 字节）：　S50：0～63；S70：0～255

输出参数：*Data：Data 为读回 16 字节数据的首地址。

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，NO_TAG_ERR，CRC_ERR，NOT_AUTH_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个。

功能描述：在验证成功后，使用该函数读 Mifare 卡中相应块的数据。Mifare 卡中一个块的数据是 16 字节，因此读写一次均是 16 个字节。

所读块号必须与之前所验证的块号在同一个扇区内，mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区。密码数据不能被读取。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）
COMMAND：0x46
LENGTH: 1
DATA[0]: Block
例如：读取块 2 的数据的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x46	0x01	0x02	0xba	0x03

TX522→主机（响应模式）：
SEQNR: 0
STATUS: OK， QUIT， NO_TAG_ERR， CRC_ERR， NOT_AUTH_ERR，PARITY_ERR，BIT_COUNT_ERR，COMM_ERR 中的某一个
LENGTH: 16
DATA[0]: 所访问块的第一个字节
：
DATA[15]: 所访问块的最后一个字节

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x10	16 字节数据	0xXX	0x03

5.3.24 带内部自动传送的值操作 1—Value

函数原型：uchar TX_Value(uchar ValueMode, uchar Block, uchar idata *Value, uchar Trans_Block);

输入参数：ValueMode：0xC0—减； 0xC1—加； 0xC2—恢复

Block：卡内块地址，对该块进行值操作，取值范围：S50：1～63；S70：1～255

*Value：4 字节数据指针，用来存储减少值或增加值，当进行恢复操作时，该值为空值。Value 是减少值或增加值存放的首地址，存放时，低地址存放高字节。

Trans_Block：传输块地址，取值范围：S50：1～63；S70：1～255。

输出参数： 无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个。

功能描述：此函数对卡内的某一块进行加、减或数据备份，该块必须为值块格式，并支持自动传送。该函数其实是 TX_Increment、TX_Decrement 或 TX_Restore 函数与 TX_Transfer 函数的组合。因此可以用该函数替换上述函数。若卡块号与传输块号相同，则将操作后的结果写入原来的块内；若卡块号与传输块号不相同，则将操作后的结果写入传输块内，结果传输块内的数据被覆盖，原块内的值不变。当模式为“恢复”时，“值”无意义。

数据块格式描述：

主机→TX522 命令模式）：
SEQNR: 0 （可自定义）
COMMAND：0x70
LENGTH: 7

DATA[0]: ValueMode
DATA[1]: Block
DATA[2]: Value(LL)
DATA[3]: Value(LH)
DATA[4]: Value(HL)
DATA[5]: Value(HH)
DATA[6]: Trans_Block
例如：将块 1 的值减 3，然后传送到块 2 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x70	0x07	0xc0 0x01 0x03 0x00 0x00 0x00 0x02	0x48	0x03

TX522→主机（响应模式）：
SEQNR: 0
STATUS: OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个
LENGTH: 0
DATA[0]: 无
例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

5.3.25 置位控制位—Set_Control_Bit

函数原型：uchar TX_Set_Control_Bit(void);
输入参数： 无
输出参数： 无
函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR 中的某一个。
功能描述：此函数设置 TX522B 模块的控制位 Ctrl（J2_1 脚）为高电平。
数据块格式描述：
主机→TX522 命令模式）：
SEQNR: 0 （可自定义）
COMMAND: 0x50
LENGTH: 0
DATA[0]: 无

TX522→主机（响应模式）：
SEQNR: 0
STATUS: OK，QUIT，COMM_ERR 中的某一个
LENGTH: 0
DATA[0]: 无

5.3.26 清除控制位—Clr_Control_Bit

函数原型：TX_Clr_Control_Bit(void);

输入参数： 无

输出参数： 无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK，QUIT，COMM_ERR 中的某一个。

功能描述：此函数设置 TX522B 模块的控制位 Ctrl（J2_1 脚）为低电平。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0 （可自定义）

COMMAND：0x51

LENGTH: 0

DATA[0]: 无

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK，QUIT，COMM_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

5.3.27 输出蜂鸣器信号—Buzzer

函数原型：TX_Buzzer(uchar Frquence, uchar Opentm, uchar Closetm, uchar Repcnt);

输入参数：Frequence: 输出方波频率，取值（0～255），对应频率（0.73～4K），0 为直流，198 对应 2K。

Opentm: 方波输出持续时间，取值（0～255），10ms 的分辨率

Closetm: 间隙时间，取值（0～255），10ms 的分辨率

Repcnt: 重复次数

输出参数： 无

函数返回：TX522B 执行命令后的状态，可能的状态值如下：OK， COMM_ERR 中的某一个。

功能描述：此函数输出一方波用于驱动交流蜂鸣器或低电平驱动直流蜂鸣器，驱动的频率、持续时间、间隙时间和重复次数可设定。下面是频率对照表。

表 20 设定值与输出频率对照表

Frequence	输出频率（KHz）	Frequence	输出频率（KHz）
0	输出低电平（直流）	140	1.333
1	0.735	160	1.515
20	0.787	180	1.74
40	0.847	198	2.00
60	0.913	200	2.04
80	0.990	220	2.50
100	1.081	240	3.17
120	1.198	255	4.00

数据块格式描述：

主机→TX522 命令模式）：

SEQNR： 0 （可自定义）
COMMAND： 0x60
LENGTH： 4
DATA[0]: Frequence
DATA[1]: Opentm
DATA[2]: Closetm
DATA[3]: Repcnt

TX522→主机（响应模式）：

SEQNR： 0
STATUS： OK， COMM_ERR 中的某一个
LENGTH： 0
DATA[0]： 无

6. 使用串口调试助手开发调试 TX522B 模块

6.1 软硬件准备

1. TX522BT+模块一块，或 TX522BT 外接 RS232 转换
2. 串口电缆
3. 5V 电源
4. 5 根连线，用于连接 TX522BT+和串口电缆、电源
5. Mifare 卡一张
6. 串口调试软件
7. 串口监视软件（选）
8. PC 机

6.2 电路连接

请按照图 6 所示连接 TX522BT 和 PC 机的串口。

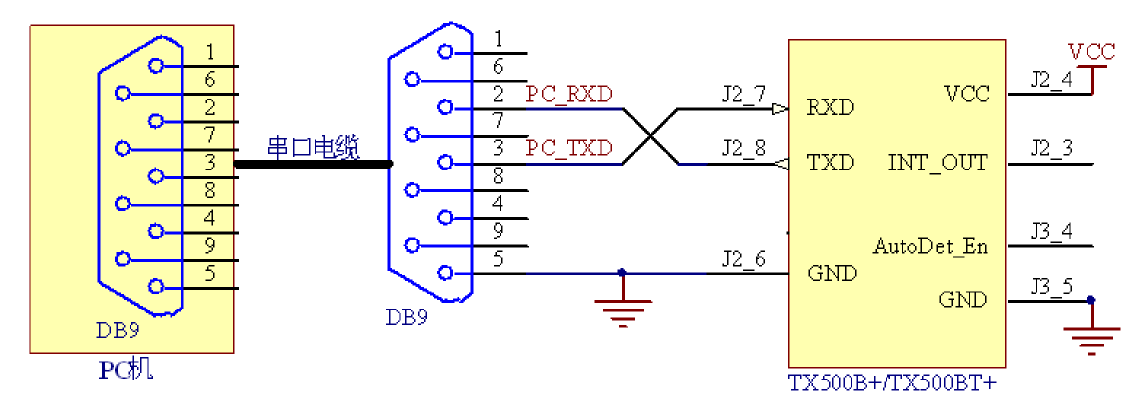


图 6 硬件接线图

6.3 设置串口调试软件

按照图 7 所示设置串口调试软件。

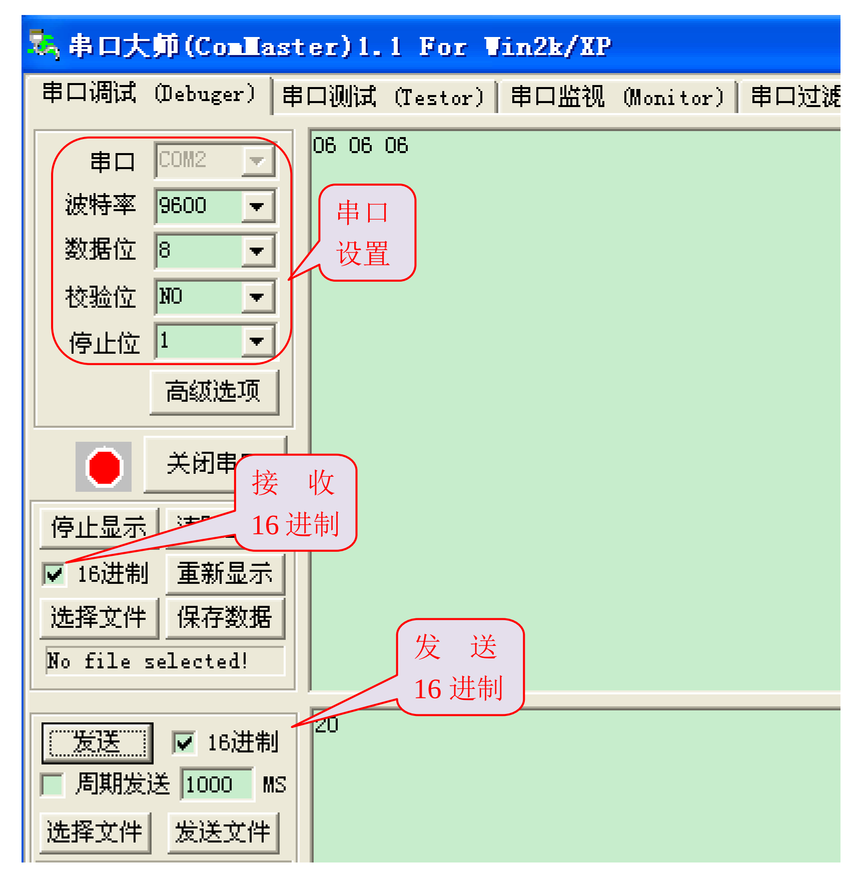


图 7 串口调试软件的设置

6.4 向 TX522BT+发送命令

发送命令的格式为：起始符(STX: 0x20) + 包序号(一般设 00 即可) + 命令字 + 数据长度 + 数据 + BCC 校验 + 结束符(EXT: 03)。详细说明见前文。

《TX522B 命令集_V1.00.txt》列出了一些常用的命令，一些可以直接拷贝到串口调试软件的发送窗口中直接进行发送。有一些需要用户进行一定的修改，例如使用的卡片不同，发送选择命令（select）时应填入相应的卡号。注意如果数据有些不同，则 BCC 校验也不同。

举例：自动检测卡片，主动回发数据，回发后继续检测，TX 全驱动，只读卡号。发送命令如图 8 所示。

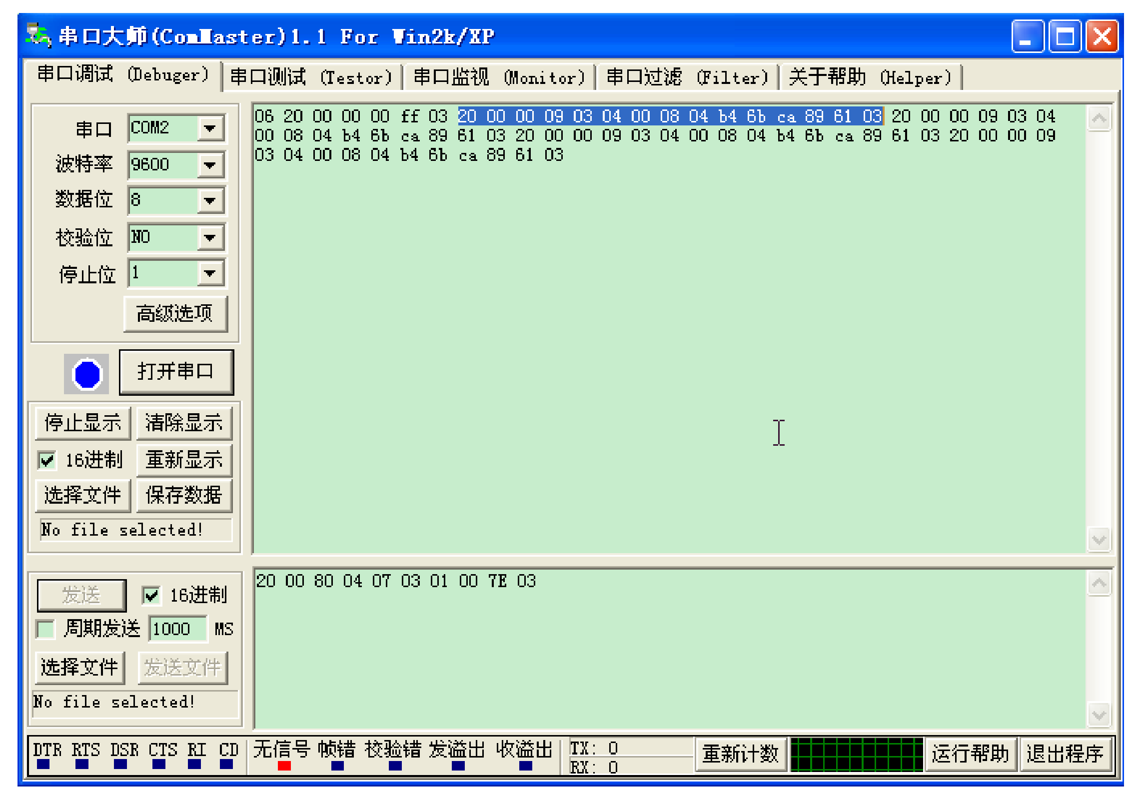


图 8 串口调试软件的设置

发送完命令后，将 mifae 卡靠近模块，串口接收区就会显示接收到的数据。图 8 接收区中的数据解释如下：

06 是主机发送 20 后模块的应答，20 00 00 00 00 ff 03 是主机发送 20 00 80 04 07 03 01 01 7E 03 命令后模块的响应。其中 20 是模块要向主机回发数据块 00 00 00 00 ff 的起始请求，等待模块的应答 06，但在这里主机并没有应答 06 给模块，因此数据块 00 00 00 00 ff 在主机先接收到 20 后的 20ms 后才会被接收到。最后一个字节 03 是本帧数据的结束符。

20 00 00 09 03 04 00 08 04 B4 6B CA 89 61 03 是自动寻卡后的发回的数据。20 是起始符，由于主机不给予应答 06，后面的数据块要 20ms 后才回发。第一个 00 是包号，最后一个字节是结束符，第二个 00 是状态表示 OK， 09 是本数据块长度， 03 天线驱动模式为全驱，04 00 是请求的应答即卡片类型，表示 S50 卡， 08 是选择的应答，表示选择 S50 卡成功， 04 是卡号长度， B4 6B CA 89 是 4 字节卡号，61 是 BCC 校验。最后一个字节 03 是本帧数据的结束符。

7. Mifare 卡工作状态介绍

Mifare 卡工作状态如表 21 所列。

表 21 Mifare 卡工作状态列表

状　态	描　　述
POWER OFF（断电）	卡片不位于读卡器有效区域，由于缺少射频磁场能量而处于断电状态，卡片不工作。
IDLE（空闲）	卡片进入读卡起有效区域内，被电磁场能量激活，延迟数毫秒后将进入 IDLE 状态。在该状态下，卡片能够解调读卡器传来的调制信号，并能对读卡器的 Request（以 IDLE 或 ALL 方式）命令进行应答，应答后返回卡片的类型。
READY（就绪）	卡片对读卡器的 Request 命令进行应答后，就进入了 READY 状态。在该状态中，可以采用比特帧防冲突算法。当卡片的唯一序列号被读卡器发来的 Selection 命令选中时，就退出本状态。
ACTIVE（激活）	当卡片的唯一序列号被读卡器选中时就进入 ACTIVE 状态。在该状态中，卡片完成本次应用所要求的全部操作。
HALT（停止）	卡片应用完成后，读卡器可通过发送 Halt 命令，使卡片进入 HALT 状态。在该状态中，卡片只对读卡器以 ALL 方式发送的 Request 命令进行应答（或被唤醒），从而又进入 READY 状态。

图 9 为 Mifare 卡的状态转换图。

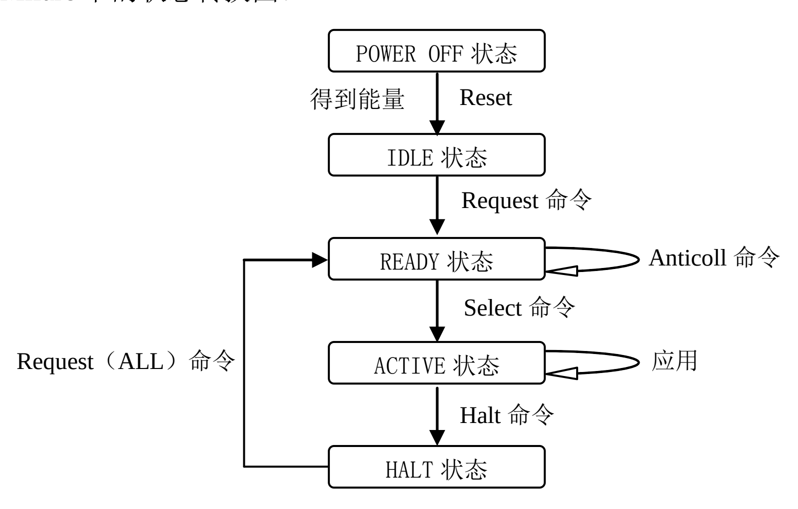


图 9 Mifare 卡状态转换示意图

8. 免责声明

- **开发预备知识**

TX®系列产品将提供尽可能全面的开发模板、驱动程序及其应用说明文档以方便用户使用，但也需要用户熟悉自己设计产品所采用的硬件平台及相关 C 语言的知识。

- **EMI 与 EMC**

TX®系列模块机械结构决定了其 EMI 性能必然与一体化电路设计有所差异。TX®系列模块的 EMI 能满足绝大部分应用场合，用户如有特殊要求，必须事先与我们协商。

TX®系列模块的 EMC 性能与用户底板的设计密切相关，尤其是电源电路、I/O 隔离、复位电路，用户在设计底板时必须充分考虑以上因素。我们将努力完善 TX®系列模块的电磁兼容特性，但不对用户最终应用产品 EMC 性能提供任何保证。

- **修改文档的权利**

东莞同欣智能保留任何时候在不事先声明的情况下对 TX®系列产品相关文档的修改权力。

- **ESD 静电放电保护**



TX®系列产品部分元器件内置 ESD 保护电路，但在使用环境恶劣的场合，依然建议用户在设计底板时提供 ESD 保护措施，特别是电源与 I/O 设计，以保证产品的稳定运行。安装 TX®系列产品，为确保安全请先将积累在身体上的静电释放，例如佩戴可靠接地的静电环，触摸接入大地的自来水管等。

9. 修订历史

版本	日期	原因
V1.00	2009/04/01	创建文档。
V1.01	2009/06/01	1. 修改典型电路原理图 2. 增加免责声明部分 3.
V1.02	2009/07/11	1. 去掉了一下与 TX500 兼容，但无用的函数 2. 固定波特率。
V1.03	2009/08/11	1. 修改了图 2，蜂鸣器和 LED 供电改为 5V，并增加续流二极管。
V1.04	2010/05/27	1. 增加了双天线配置。
V1.05	2010/07/30	1. 增加分体式模块的管脚描述

10. 销售信息

同欣智能
地　址：广东省东莞市石碣镇沙腰管理区林屋洲
邮　编：523292
电　话：+86 (0769) 86019853; 15916881410
传　真：+86 (0769) 86019852
网　　址：http:// www.TXRFID.com
E-mail：sales@TXRFID.com　support@TXRFID.com